

# – AZ 1 –

## – Projekt zaliczeniowy –

## – Dokumentacja –

Gołębiewski Bartłomiej

IZ07TC1

17678

WIT

### Spis treści

1. Obsługa skryptu.....	2
2. Konta użytkownika.....	3
a) Tworzenie konta użytkownika.....	3
b) Tworzenie wielu kont na podstawie CSV .....	4
c) Blokowanie konta użytkownika.....	6
d) Zmiana hasła konta użytkownika .....	6
3. Obsługa kont grup .....	7
a) Tworzenie nowych grup .....	7
b) Dodawanie użytkowników do grup.....	8
4. Raporty .....	8
a) Lista grup z członkami.....	9
b) Lista zablokowanych kont w domenie .....	9
c) Lista szczegółowych informacji o kontach użytkowników .....	10
d) Lista szczegółowych informacji o kontach komputerów w domenie.....	10
e) Lista jednostek organizacyjnych w domenie .....	11
f) Informacje o domenie .....	11
Spis ilustracji :	11

## 1. Obsługa skryptu.

Plikiem głównym zawierającym wszystkie niezbędne funkcje do działania jest „AZ1\_Projekt\_Bartłomiej\_Golebiewski\_IZ07TC1\_17678.ps1”.

```
AZ1_Projekt_Bartłomiej_Golebiewski_IZ07TC1_17678.ps1 X
1  #Bartłomiej Gołębiewski IZ07TC1 17678 AZ1_Projekt
2  #####
3  #Zbiór funkcji potrzebnych do działania
4  #####
5
6  #funkcja tworząca konto użytkownika AD
7  function Dodawanie-Usera
8  {
9      ...
10 }
11
12 #funkcja tworząca wiele kont użytkownika AD na podstawie uzytkownicy.csv
13 function DodawanieWielu-Userow
14 {
15     ...
16 }
17
18 #funkcja blokująca konta
19 function Blokada-Usera
20 {
21     ...
22 }
23
24 #funkcja zmieniająca hasło
25 function Zmiana-Hasla
26 {
27     ...
28 }
29
30 #funkcja tworząca nową grupę
31 function Nowa-Grupa
32 {
33     ...
34 }
35
36 #funkcja dodająca konto do grupy
37 function DodajDo-Grupy
38 {
39     ...
40 }
41
42 #funkcja tworząca listę grup
43 function Lista-Grup
44 {
45     ...
46 }
47
48 #funkcja tworząca listę zablokowanych kont
49 function ListaZablokowanych-Kont
50 {
51     ...
52 }
53
54 #funkcja tworząca listę kont
55 function Lista-Kont
56 {
57     ...
58 }
59
60 #funkcja tworząca listę jednostek org.
61 function Lista-JOrg
62 {
63     ...
64 }
65
66 #funkcja wypisująca informacje o domenie
67 function Info-Domena
68 {
69     ...
70 }
71
72 #funkcja tworząca listę komputerów
73 function Lista-Komputerow
74 {
75     ...
76 }
77
78 #####
79 #Menu
80 #####
81 cls
82
```

Rysunek 1. Cały skrypt

Może on być uruchamiany w dowolnym miejscu w domenie za pomocą prawego kliku myszka i wyboru opcji „Run with Powershell” bądź przez „Windows PowerShell ISE” używając przycisku „F5”(Run Script)., skrypt sam generuje sobie pliki i tworzy folderu. Po jego uruchomieniu powita nas okno startowe.

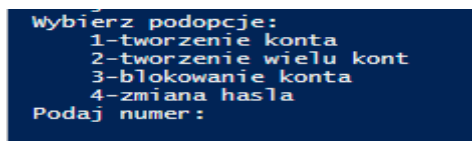
```
Wybierz opcje z menu:
1-konta użytkownika
2-grupy
3-raporty
Podaj numer:
```

Rysunek 2. Główne menu

## 2. Konta użytkownika.

### a) Tworzenie konta użytkownika

Po wybraniu opcji „1-konta użytkownika” skrypt wyświetli nowe okno z następującymi opcjami.



```
Wybierz podopcje:
1-tworzenie konta
2-tworzenie wielu kont
3-blokowanie konta
4-zmiana hasla
Podaj numer:
```

Rysunek 3. Menu tworzenia kont

Za opcje „1-tworzenie konta” odpowiada funkcja **Dodawanie-Usera** :

```
#funkcja tworząca konto użytkownika AD
function Dodawanie-Usera
{
    Write-Host "Tworzenie nowego uzytkownika"
    $imie = Read-Host "Podaj Imie"
    $nazwisko = Read-Host "Podaj Nazwisko"
    $dzial = Read-Host "Podaj Dzial"
    #utworzenie loginu na podstawie imienia i nazwiska
    $login = $imie+"."+ $nazwisko
    #znalezienie wolnego loginu
    $zapamietaj = $login
    $i = 1
    While(Get-ADUser -Filter "sAMAccountName -eq '$($login)'" )
    {
        $login = $zapamietaj
        $login = $login+$i
        $i++
    }
    #wyciagniecie informacji o domenie
    $infodomena = Get-ADDomain
    $domena = $infodomena.Forest
    #utworzenie maila na podstawie loginu i domeny
    $email = $login+"@"+$domena
    #utworzenie nazwy na podstawie imienia i nazwiska
    $nazwa = $imie+" " + $nazwisko
    $i=1
    $zapamietaj=$nazwa
    While(Get-ADUser -Filter "Name -eq '$($nazwa)'" )
    {
        $nazwa = $zapamietaj
        $nazwa = $nazwa+$i
        $i++
    }
    #generowanie hasla
    $haslo = ""
    for($i = 1; $i -le 10; $i++)
    {
        $haslo += [char](Get-Random -Minimum 48 -Maximum 122)
    }
    #konwertowanie hasla
    $hasloAD = ConvertTo-SecureString $haslo -AsPlainText -Force

    #tworzenie katalogu dla plikow z haslami uzytkownikow
    $sciezka = '.\Konta-haslo'
    if(!(Test-Path -Path $sciezka))
    {
        New-Item -ItemType "directory" -Path $sciezka
    }
    #tworzenie pliku z haslem uzytkownika
    $plik = "17678_ $nazwa.csv"
    New-Item -ItemType "file" -Name $plik -Path $sciezka
    "$login,$haslo"|Out-File -FilePath "$sciezka\$plik"

    #wskazanie jednostki organizacyjnej na konta
    $sdn = $infodomena.DistinguishedName
    $ou = "OU=Projekt,"+$sdn
    #sprawdzenie czy ou istnieje
    if (!(Get-ADOrganizationalUnit -Filter "distinguishedName -eq '$ou'"))
    {
        New-ADOrganizationalUnit -Name "Projekt" -Path $sdn
    }
    #tworzenie uzytkownika AD
    New-ADUser -DisplayName: $nazwa -GivenName: $imie -Surname:$nazwisko -Name: $nazwa -Department: $dzial
    #pobranie daty i informacji kto
    $kto = whoami
    $kiedy = Get-Date
}
```

```

#tworzenie katalogu dla plikow z logami
$sciezka = '..\Logi'
if(!(Test-Path -Path $sciezka))
{
    New-Item -ItemType "directory" -Path $sciezka
}
#tworzenie pliku z logami dotyczace tworzenia konta uzytkownika
$sciezka = '..\Logi\17678-create_user.csv'
{
    New-Item -ItemType "file" -Path '..\Logi\' -Name '17678-create_user.csv'
    "kto,kiedy,konto" | Out-File -FilePath $sciezka -Encoding utf8
}
"$kto,$kiedy,$login" | Out-File -FilePath $sciezka -Encoding utf8 -Append
}

```

Rysunek 4. Funkcja tworząca konta

Która pobiera parametry, po czym tworzy i sprawdza login, wyciąga informacje o domenie, tworzy maila i nazwe, generuje i konwertuje hasło oraz tworzy katalogi dla plików z hasłami i logami.

## b) Tworzenie wielu kont na podstawie CSV

Za opcję „2-tworzenie wielu kont” odpowiada funkcja **DodawanieWielu-Userow** :

```

#funkcja tworząca wiele kont uzytkownika AD na podstawie uzytkownicy.csv
function DodawanieWielu-Userow
{
    Write-Host "Tworzenie wielu uzytkownikow"
    #tworzenie pustego pliku csv do uzupelnienia
    $sciezka = "..\uzytkownicy.csv"
    if(!(Test-Path -Path $sciezka))
    {
        New-Item -ItemType "file" -Name "uzytkownicy.csv" -Path "..\"
        "imie,nazwisko,dzial" | Out-File -FilePath $sciezka -Append -Encoding utf8
        Write-Host "Utworzono plik uzytkownicy.csv w lokalizacji C:\Projekt."
    }
    else
    {
        #wczytanie pliku csv
        $csv = Import-Csv -Path "..\uzytkownicy.csv" -Encoding UTF8 -Delimiter ","
        #petla tworząca kolejnych uzytkownikow
        foreach($konto in $csv)
        {
            #pobranie danych z pliku csv
            $imie = $konto.imie
            $nazwisko = $konto.nazwisko
            $dzial = $konto.dzial

            $login = $imie+"."+ $nazwisko
            $zapamietaj = $login
            $i = 1
            While(Get-ADUser -Filter "sAMAccountName -eq '$($login)'" )
            {
                $login = $zapamietaj
                $login = $login+$i
                $i++
            }
            #wyciągnięcie informacji o domenie
            $infodomena = Get-ADDomain
            $domena = $infodomena.Forest
            #utworzenie maila na podstawie loginu i domeny
            $email = $login+"@"+$domena
            #utworzenie nazwy na podstawie imienia i nazwiska
            $nazwa = $imie+" " + $nazwisko
            $i=1
            $zapamietaj=$nazwa
            While(Get-ADUser -Filter "Name -eq '$($nazwa)'" )
            {
                $nazwa = $zapamietaj
                $nazwa = $nazwa+$i
                $i++
            }
        }
    }
}

```

```

#generowanie hasła
$hasło = ""
for($i = 1; $i -le 10; $i++)
{
    $hasło += [char](Get-Random -Minimum 48 -Maximum 122)
}
#konwertowanie hasła
$hasłoAD = ConvertTo-SecureString $hasło -AsPlainText -Force

#tworzenie katalogu dla plików z hasłami użytkowników
$sciezka = '.\Konta-hasło'
if(!(Test-Path -Path $sciezka))
{
    New-Item -ItemType "directory" -Path $sciezka
}
#tworzenie pliku z hasłem użytkownika
$plik = "17678_$nazwa.csv"
New-Item -ItemType "file" -Name $plik -Path $sciezka
"$login,$hasło" | Out-File -FilePath "$sciezka\$plik"

#wskazanie jednostki organizacyjnej na konta
$dn = $infodomena.DistinguishedName
$sou = "OU=Projekt,"+$dn
#sprawdzenie czy ou istnieje
if (!(Get-ADOrganizationalUnit -Filter "distinguishedName -eq '$sou'"))
{
    New-ADOrganizationalUnit -Name "Projekt" -Path $dn
}
#tworzenie użytkownika AD
New-ADUser -DisplayName: $nazwa -GivenName: $imie -Surname:$nazwisko -Name: $nazwa -Department: $dział -Path:
#pobranie daty i informacji kto
$któ = whoami
$ktedy = Get-Date
#tworzenie katalogu dla plików z logami
$sciezka = '.\Logi'
if(!(Test-Path -Path $sciezka))
{
    New-Item -ItemType "directory" -Path $sciezka
}
#tworzenie pliku z logami dotyczące tworzenia konta użytkownika
$sciezka = '.\Logi\17678-create_user.csv'
if(!(Test-Path -Path $sciezka))
{
    New-Item -ItemType "file" -Path '.\Logi\' -Name '17678-create_user.csv'
    "kto,ktedy,konto" | Out-File -FilePath $sciezka -Encoding utf8
}
"$któ,$ktedy,$login" | Out-File -FilePath $sciezka -Encoding utf8 -Append
}
}

```

Rysunek 5. Funkcja tworząca wiele kont

Która najpierw tworzy plik csv z którego pobiera później parametry do tworzonego konta, generuje i konwertuje hasło, tworzy logi i katalogi w których owe będą przechowywane.

### c) Blokowanie konta użytkownika

Za opcję „3-blokowanie usera” odpowiada funkcja **Blokada-Usera** :

```
function Blokada-Usera
{
    #pobranie loginu od uzytkownika
    Write-Host "Blokowanie konta"
    $login = Read-Host "Podaj login"
    #wylaczenie konta
    Set-ADUser -Identity $login -Enabled $false
    #pobranie daty i informacji kto
    $data = Get-Date -Format("dd.MM.yyyy")
    $kto = whoami
    #tworzenie katalogu dla plikow z logami
    $sciezka = '.\Logi'
    if(!(Test-Path -Path $sciezka))
    {
        New-Item -ItemType "directory" -Path $sciezka
    }
    #tworzenie pliku z logami dotyczace tworzenia konta uzytkownika
    $sciezka = '.\Logi\17678-zablokowane_konta-$data.csv'
    if(!(Test-Path -Path $sciezka))
    {
        New-Item -ItemType "file" -Path '.\Logi\' -Name '17678-zablokowane_konta-$data.csv'
        "kto,kiedy,konto" | Out-File -FilePath $sciezka -Encoding utf8
    }
    "$kto,$kiedy,$login" | Out-File -FilePath $sciezka -Encoding utf8 -Append
}
```

Rysunek 6. Funkcja blokująca

Która pobiera login jako parametr a następnie blokuje wcześniej podane konto, tworząc jednocześnie logi informujące kto dokonał blokady oraz pliki w których owe się znajdują.

### d) Zmiana hasła konta użytkownika

Za opcję „4-zmiana hasła” odpowiada funkcja **Zmiana-Hasła** :

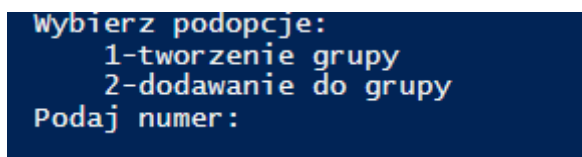
```
#funkcja zmieniajaca haslo
function Zmiana-Hasla
{
    Write-Host "Zmienianie hasla"
    #pobranie loginu i hasla od uzytkownika
    $login = Read-Host "Podaj login"
    $haslo = Read-Host "Podaj haslo"
    #konwertowanie hasla
    $hasloAD = ConvertTo-SecureString $haslo -AsPlainText -Force
    #zmiana hasla uzytkownika
    Set-ADAccountPassword -Identity $login -NewPassword $hasloAD -Reset:$true
    #pobranie daty i informacji kto
    $kiedy = Get-Date -Format("dd-MM-yyyy")
    $kto = whoami
    #tworzenie katalogu dla plikow z logami
    $sciezka = '.\Logi'
    if(!(Test-Path -Path $sciezka))
    {
        New-Item -ItemType "directory" -Path $sciezka
    }
    #tworzenie pliku z logami dotyczace tworzenia konta uzytkownika
    $sciezka = ".\Logi\17678-zmiana_hasla-$kiedy.csv"
    if(!(Test-Path -Path $sciezka))
    {
        New-Item -ItemType "file" -Path ".\Logi\' -Name "17678-zmiana_hasla-$kiedy.csv"
        "kto,kiedy,konto" | Out-File -FilePath $sciezka -Encoding utf8
    }
    "$kto,$kiedy,$login" | Out-File -FilePath $sciezka -Encoding utf8 -Append
}
```

Rysunek 7. Funkcja zmieniająca

Której zadaniem jest pobranie loginu i hasła jako parametrów, zmiana hasła oraz wygenerowanie folderów wraz z logami informującymi o tym kto dokonał zmian.

### 3. Obsługa kont grup

Po wybraniu opcji „2-tworzenie grup” skrypt wyświetli nowe okno z następującymi opcjami.



```
Wybierz podopcje:
    1-tworzenie grupy
    2-dodawanie do grupy
Podaj numer:
```

Rysunek 8. Menu grup

#### a) Tworzenie nowych grup

Za opcje „1-tworzenie grupy” odpowiada funkcja **Nowa-Grupa** :

```
#funkcja tworząca nowa grupe
function Nowa-Grupa
{
    #pobranie loginu i hasła od użytkownika
    Write-Host "Tworzenie nowej grupy"
    $grupa = Read-Host "Podaj nazwę grupy"
    if(Get-ADGroup -Filter "sAMAccountName -eq '$($grupa)'" )
    {
        Write-Host "Grupa już istnieje"
    }
    else
    {
        $infodomena = Get-ADDomain
        $dn = $infodomena.DistinguishedName

        $ou = "OU=Projekt,"+$dn
        #sprawdzenie czy ou istnieje
        if (!(Get-ADOrganizationalUnit -Filter "distinguishedName -eq '$ou'"))
        {
            New-ADOrganizationalUnit -Name "Projekt" -Path $dn
        }
        New-ADGroup -Name $grupa -DisplayName $grupa -SamAccountName $grupa -Path $ou -GroupCategory Security -GroupScope Global
        #pobranie daty i informacji kto
        $kiedy = Get-Date
        $kto = whoami
        #tworzenie katalogu dla plików z logami
        $sciezka = ".\Logi"
        if(!(Test-Path -Path $sciezka))
        {
            New-Item -ItemType "directory" -Path $sciezka
        }
        #tworzenie pliku z logami dotyczące tworzenia konta użytkownika
        $sciezka = ".\Logi\17678-create_group.csv"
        if(!(Test-Path -Path $sciezka))
        {
            New-Item -ItemType "file" -Path ".\Logi\" -Name '17678-create_group.csv'
            "kto,kiedy,grupa" | Out-File -FilePath $sciezka -Encoding utf8
        }
        "$kto,$kiedy,$grupa" | Out-File -FilePath $sciezka -Encoding utf8 -Append
    }
}
```

Która pobiera parametr nazwy grupy, sprawdza czy istnieje, jeśli nie to ją tworzy razem z plikiem i logami zawierającymi informacje o tym kto dokonał zmian.

Rysunek 9. Funkcja tworząca nowe grupy

## b) Dodawanie użytkowników do grup

Za opcje „2-dodawanie do grupy” odpowiada funkcja **DodajDo-Grupy** :

```
#funkcja dodajaca konto do grupy
function DodajDo-Grupy
{
    #pobranie loginu i hasla od uzytkownika
    Write-Host "Dodawanie do grupy"
    $grupa = Read-Host "Podaj nazwe grupy"
    $login = Read-Host "Podaj login uzytkownika"
    #dodanie do grupy
    Add-ADGroupMember -Identity $grupa -Members $login
    #pobranie daty i informacji kto
    $kiedy = Get-Date
    $kto = whoami
    #tworzenie katalogu dla plikow z logami
    $sciezka = '.\Logi'
    if(!(Test-Path -Path $sciezka))
    {
        New-Item -ItemType "directory" -Path $sciezka
    }
    #tworzenie pliku z logami dotyczace tworzenia konta uzytkownika
    $sciezka = '.\Logi\17678-zmiana_czlonkostwa_grup.txt'
    if(!(Test-Path -Path $sciezka))
    {
        New-Item -ItemType "file" -Path '.\Logi\' -Name '17678-zmiana_czlonkostwa_grup.txt'
        "kto,kiedy,grupa,login" | Out-File -FilePath $sciezka -Encoding utf8
    }
    "$kto,$kiedy,$grupa,$login" | Out-File -FilePath $sciezka -Encoding utf8 -Append
}
```

Rysunek 10. Funkcja dodawania użytkowników

Której zadaniem jest pobranie nazwy grupy i loginu jako parametru, po czym dodaje konto do grupy i pobiera jako parametr informacje kto dokonał zmian tworząc plik z logami.

## 4. Raporty

Po wybraniu opcji „3-raporty” skrypt wyświetli nowe okno z następującymi opcjami.

```
Wybierz podopcje:
1-lista grup
2-lista zablokowanych kont
3-lista informacji o uzytkownikach
4-lista informacji o komputerach
5-lista jednostek organizacyjnych
6-lista informacji o domenie
Podaj numer :
```

Rysunek 11. Menu raporty



## a) Lista grup z członkami

Za opcje „1-lista grup” odpowiada funkcja **Lista-Grup** :

```
#funkcja tworząca listę grup
function Lista-Grup
{
    Write-Host "Tworzenie listy grup"
    #pobranie wszystkich grup z AD
    $lista = Get-ADGroup -Filter *
    $lista = $lista.SamAccountName

    foreach($grupa in $lista)
    {
        #znalezienie członków grupy z listy
        $członkowie = Get-ADGroupMember -Identity $grupa
        $członkowie = $członkowie.SamAccountName
        #tworzenie katalogu dla plików z członkami grup
        $sciezka = '.\Grupy'
        if(!(Test-Path -Path $sciezka))
        {
            New-Item -ItemType "directory" -Path $sciezka
        }
        #tworzenie pliku z logami dotyczące tworzenia konta użytkownika
        $sciezka = ".\Grupy\17678-$grupa.txt"
        if(!(Test-Path -Path $sciezka))
        {
            New-Item -ItemType "file" -Path '.\Logi\' -Name "17678-$grupa.txt"
        }
        "$członkowie" | Out-File -FilePath $sciezka -Encoding utf8
    }
}
```

Rysunek 12. Funkcja tworząca listę grup

Która pobiera wszystkie grupy z AD po czym tworzy katalog z plikami tekstowymi zawierającymi dane o grupach.

## b) Lista zablokowanych kont w domenie

Za opcje „2-lista zablokowanych kont” odpowiada funkcja **ListaZablokowanych-Kont** :

```
#funkcja tworząca listę zablokowanych kont
function ListaZablokowanych-Kont
{
    Write-Host "Tworzenie listy zablokowanych kont"
    #pobranie wyłączonych kont użytkowników
    $konta = Get-ADUser -Properties whenChanged -Filter {Enabled -eq $False} | select Name, DistinguishedName, SID, whenChanged
    #tworzenie katalogu dla plików z zablokowanymi kontami
    $sciezka = '.\Konta-zablokowane'
    if(!(Test-Path -Path $sciezka))
    {
        New-Item -ItemType "directory" -Path $sciezka
    }
    #tworzenie pliku z logami dotyczące tworzenia konta użytkownika
    $sciezka = '.\Konta-zablokowane\17678-zablokowane_konta.csv'
    $konta | Export-Csv -Path $sciezka -NoTypeInformation -Encoding UTF8
}
```

Rysunek 13. Funkcja tworząca listę zablokowanych kont

Która pobiera jako parametr zablokowane konta i tworzy katalog z plikami zawierającymi informacje o zablokowanych kontach.

### c) Lista szczegółowych informacji o kontach użytkowników

Za opcje „3-lista zablokowanych kont” odpowiada funkcja **Lista-Kont** :

```
#funkcja tworząca listę kont
function Lista-Kont
{
    Write-Host "Tworzenie listy kont"
    #pobranie wszystkich kont użytkowników
    $konta = Get-ADUser -Properties * -Filter * | select GivenName, SurName, UserPrincipalName, SamAccountName, DistinguishedName, whenCreated,
    whenChanged, LastLogon, PasswordLastSet
    #tworzenie katalogu dla plików z kontami
    $sciezka = '.\Konta'
    if(!(Test-Path -Path $sciezka))
    {
        New-Item -ItemType "directory" -Path $sciezka
    }
    #tworzenie pliku z logami dotyczące tworzenia konta użytkownika
    $sciezka = '.\Konta\17678-uzytkownicy.csv'
    $konta | Export-Csv -Path $sciezka -NoTypeInformation -Encoding UTF8
}
```

Rysunek 14. Funkcja tworząca listę szczegółowych informacji o kontach użytkowników

Pobiera wszystkie konta po czym tworzy katalogi z plikami zawierającymi informacje o kontach.

### d) Lista szczegółowych informacji o kontach komputerów w domenie

Za opcje „4-lista informacji o komputerach” odpowiada funkcja **Lista-Komputerow** :

```
#funkcja tworząca listę komputerow
function Lista-Komputerow
{
    Write-Host "Tworzenie listy komputerow."
    #pobranie wszystkich komputerow
    $komputery = Get-ADComputer -Filter * -Properties * | Select-Object name, operatingSystem, SID,
    DistinguishedName, Enabled, PasswordLastSet, Created
    #stworzenie unikatowej listy systemow
    $systemy = $komputery.operatingSystem | select -Unique
    #tworzenie katalogu dla komputerow
    $sciezka = '.\Komputery'
    if(!(Test-Path -Path $sciezka))
    {
        New-Item -ItemType "directory" -Path $sciezka
    }

    ForEach($system in $systemy)
    {
        $komputer = $komputery | Where-Object {$_.operatingSystem -eq $system}
        #wyciągnięcie informacji o domenie
        $infodomena = Get-ADDomain
        $domena = $infodomena.Forest
        $sciezka = ".\Komputery\17678-$domena-$system.csv"
        $komputer | Export-Csv -Path $sciezka -Encoding utf8 -NoTypeInformation
    }
}
```

Rysunek 15. Funkcja tworząca listę informacji o kontach komputerów

Której zadaniem jest pobranie wszystkich jednostek organizacyjnych po czym utworzenie katalogu z logami dotyczącymi tworzenia konta.

## e) Lista jednostek organizacyjnych w domenie

Za opcje „5-lista jednostek organizacyjnych” odpowiada funkcja **Lista-JOrg** :

```
#funkcja tworząca listę jednostek org.
function Lista-JOrg
{
    Write-Host "Tworzenie listy jednostek org."
    #pobranie wszystkich jednostek org
    $jednostki = Get-ADOrganizationalUnit -Filter * | select Name, DistinguishedName | Sort-Object DistinguishedName
    #tworzenie katalogu dla plików z j. org.
    $sciezka = '.\Jednostki'
    if(!(Test-Path -Path $sciezka))
    {
        New-Item -ItemType "directory" -Path $sciezka
    }
    #tworzenie pliku z logami dotyczące tworzenia konta użytkownika
    $sciezka = '.\Jednostki\17678-jednostki.csv'
    $jednostki | Export-Csv -Path $sciezka -NoTypeInformation -Encoding UTF8
}
```

Rysunek 16. Funkcja tworząca listę jednostek organizacyjnych.

Która pobiera wybrane informacje o domenie i tworzy plik zawierający logi z tymi informacjami.

## f) Informacje o domenie

Za opcje „6-lista informacji o domenie” odpowiada funkcja **Info-Domena** :

```
#funkcja wypisująca informacje o domenie
function Info-Domena
{
    Write-Host "Wypisywanie informacji o domenie"
    #pobranie informacji o domenie
    $domena = Get-ADDomain | select Name, NetBIOSName, DomainMode, DomainSID, ComputersContainer, UsersContainer
    #tworzenie katalogu dla domeny
    $sciezka = '.\Domena'
    if(!(Test-Path -Path $sciezka))
    {
        New-Item -ItemType "directory" -Path $sciezka
    }
    #tworzenie pliku z logami dotyczące tworzenia konta użytkownika
    $sciezka = '.\Domena\17678-domain_info.csv'
    $domena | Export-Csv -Path $sciezka -NoTypeInformation -Encoding UTF8
}
```

Rysunek 17. Funkcja tworząca listę informacji o domenie

Która pobiera wybrane informacje o domenie i tworzy plik zawierający logi z tymi informacjami.

## Spis ilustracji :

Rysunek 1. Cały skrypt.....	2
Rysunek 2. Główne menu.....	2
Rysunek 3. Menu tworzenia kont.....	3
Rysunek 4. Funkcja tworząca konta .....	4
Rysunek 5. Funkcja tworząca wiele kont.....	5
Rysunek 6. Funkcja blokująca .....	6
Rysunek 7. Funkcja zmieniająca .....	6
Rysunek 8. Menu grup.....	7
Rysunek 9. Funkcja tworząca nowe grupy .....	7
Rysunek 10. Funkcja dodawania użytkowników .....	8
Rysunek 11. Menu raporty .....	8
Rysunek 12. Funkcja tworząca listę grup.....	9
Rysunek 13. Funkcja tworząca listę zablokowanych kont.....	9
Rysunek 14. Funkcja tworząca listę szczegółowych informacji o kontach użytkowników.....	10
Rysunek 15. Funkcja tworząca listę informacji o kontach komputerów .....	10
Rysunek 16. Funkcja tworząca listę jednostek organizacyjnych. ....	11
Rysunek 17. Funkcja tworząca listę informacji o domenie .....	11